Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

Trip-Wise Driver Behavior Analysis and Vehicle Health Recommendation System using Machine Learning

Yatharth Singhai
PG Scholar, Department of Computer Engineering
Shri G.S. Institute of Technology and Science
Indore - 452001, Madhya Pradesh, India

Prof. D.A. Mehta
Professor, Department of Computer Engineering
Shri G.S. Institute of Technology and Science
Indore - 452001, Madhya Pradesh, India

***Abstract*—This study presents a comprehensive machine learning approach for driver behavior classification using vehicle telematics data from multiple diverse datasets. A robust data processing pipeline was developed to handle multi-format data sources, including intelligent trip boundary detection, dynamic user management, and comprehensive data quality control. The system addresses heterogeneous data challenges through advanced column standardization and automated data validation techniques. Multiple machine learning algorithms were systematically evaluated. Logistic Regression achieved the best performance with 90.0% accuracy and 0.901 weighted F1-score, outperforming Support Vector Machines (86.7% accuracy), while Random Forest and XGBoost achieved 66.7% accuracy each. Analysis of model coefficients showed that speed variability, maximum RPM, acceleration variability, brake events, and RPM variability are among the most influential predictors when distinguishing between Safe, Moderate, and Aggressive trips. The research addresses critical data quality challenges in post-trip trip-wise telematics applications while demonstrating methodological rigor through prevention of data leakage in behavioral classification models. The proposed framework establishes best practices for multi-source vehicle data integration and provides a foundation for practical driver assessment systems in fleet management and usage-based insurance applications. Features used exclusively for rule-based scoring and feedback generation were explicitly excluded from machine learning training to prevent information leakage.**

***Index Terms*—Driving behavior, vehicle safety, telematics, machine learning, K-Means clustering, Logistic Regression, feature engineering, post-trip analysis, trip-wise scoring.**

I. Introduction

Driving behaviour forms part of the definitive direction in road safety, vehicle performance, and environmental sustainability. Traditional driving techniques such as harsh braking, forceful acceleration, and excessive speeding not only increase the risk of accidents but also lead to higher fuel consumption and destructive mechanical wear. According to the World Health Organization (WHO), over 90% of traffic accidents can be attributed to human error, highlighting the necessity for interventions that specifically target driver behaviour [1].

Recent advancements in car telematics, on-board diagnostics (OBD-II), and Internet-of-Things (IoT) based data capture systems have made it possible to gather extensive vehicular and driver-related information [2], [3]. Nevertheless, several obstacles remain in the process of deriving actionable insights that are specific and relevant to individual drivers. One of the key limitations is *data fragmentation*, where essential trip metrics—such as speed, fuel consumption, engine RPM, and braking events—are dispersed across different platforms and proprietary systems. This makes unified analysis and interpretation difficult.

Furthermore, existing solutions often lack personalization. They do not incorporate a driver's historical behavioural patterns into feedback generation. Additionally, the high cost and infrastructure requirements of many commercial telematics platforms limit their accessibility to individual drivers, small-scale researchers, and fleet operators.

This study introduces an open-source, modular *Vehicle Trip Analysis Dashboard* to address these challenges using a post-trip analysis paradigm. The primary objectives of the proposed system are:

1) To provide a framework for an interactive, web-based system where trip-level data (e.g., speed, RPM, fuel consumption, and brake patterns) can be viewed after each completed trip (post-trip) using graphical and adjustable components;
2) To design and integrate a machine learning pipeline that uses K-Means clustering (k = 3) for unsupervised behavior discovery and multi-class Logistic Regression as the final classifier for trip-wise driving behaviour;
3) To offer a tailored user experience including secure login, personalized trip analytics, and trend monitoring over sequential time intervals;

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

4) To ensure scalability through lightweight computational requirements and open-source technologies such as Flask (backend), SQLite (data store), and Chart.js (visualization).

The proposed system is intended to equip drivers and stakeholders with practical, actionable insights that encourage safer, more cost-effective, and environmentally responsible driving practices.

II. Literature Review

The analysis of driving behavior has significantly evolved, embracing the use of telematics, on-board diagnostics (OBD-II), and a wide-scale application of machine learning (ML) techniques. This literature review explores emerging trends in the field, characterizes major commercial platforms, and highlights current research gaps to inform this investigation.

A. Data Collection Techniques

Modern driving analytics relies on a variety of data sources, including Controller Area Network (CAN) bus signals, OBD-II interfaces, GPS data, and increasingly, smartphone and Internet of Things (IoT) sensors. As evidenced by Fugiglando et al. [3], CAN bus and OBD-II data provide fine-grained behavioral parameters that are effective for detecting aggressive driving. Similarly, Rizbood et al. [4] demonstrate that smartphone sensors—such as accelerometers and GPS—can yield scalable, context-rich driving signals.

B. Machine Learning and Data Mining Approaches

Various machine learning techniques have been utilized for driving pattern recognition. Algorithms such as Random Forests and Support Vector Machines (SVMs) remain attractive due to their interpretability and classification robustness. According to Garefalakis et al. [5], Random Forest models are particularly effective in detecting risky driving behaviors from real-world data.

Deep learning techniques have also been explored extensively. Jain and Mittal [6] employed LSTM autoencoders to detect time-dependent driving behavior, while Kwon et al. [7] demonstrated high accuracy using deep CNN-LSTM networks. Unsupervised methods such as k-means clustering are instrumental in identifying driver-style archetypes from unlabeled data [3]. Additionally, reinforcement learning is increasingly used for adaptive driving policy optimization, particularly in Advanced Driver-Assistance Systems (ADAS) and autonomous vehicle systems [9], [10].

C. Applications in Industry and Insurance

Commercial fleet management systems like Frotcom, Geotab, and Samsara have effectively reduced fuel consumption and accident rates by integrating telematics with driver behavior scoring dashboards. For instance, Frotcom reported a 7% reduction in fuel costs and a 70% decline in accident occurrences following dashboard implementation [13].

Usage-Based Insurance (UBI) systems are increasingly shifting toward behavior-based risk evaluation. Arumugam et al. [14] highlight the incorporation of ML-based risk scoring for real-time policy customization in the insurance industry.

D. Ethical and Human Factors

Recent studies have emphasized critical ethical concerns such as data privacy, informed consent, algorithmic fairness, and human-AI interaction. Research by Zylius et al. and Liao et al. underscores the necessity of collecting privacy-preserving vehicle data and designing transparent, user-friendly AI feedback systems [11], [12]. Beyond predictive accuracy, system effectiveness also hinges on driver acceptance and the context-aware delivery of system recommendations.

E. Current Research Gaps

Despite significant advancements, existing systems exhibit several limitations:

- Lack of open-source, modular, and easily extendable frameworks, restricting adoption by independent drivers, academic researchers, and small fleet operators;
- Inadequate support for interpretable, personalized driver scoring and actionable user feedback;
- Insufficient consideration of ethical, privacy, and contextual adaptability concerns in commercial deployments.

III. Methodology

The proposed Vehicle Trip Analysis Dashboard was developed using a modular and structured methodology. This approach ensures scalability, maintainability, and user-centric design. The complete pipeline encompasses architectural design, data preparation, feature engineering, model training, and interactive visual feedback.

A. System Architecture and Modular Design

The system is composed of loosely coupled layers, allowing independent development and testing of each module:

- **User Management:** Secure registration, login, and session control using Flask-Login.
- **Trip Data Collection:** Supports both real-world telematics data and synthetic trip simulation including parameters like speed, RPM, fuel, brake events, and GPS.
- **ML Model Integration:** Uses K-Means clustering (k = 3) to derive initial behavior labels and Logistic Regression as the final deployed classifier, with extensible support for other models. Backend ML logic is decoupled from the UI.
- **Visualization Engine:** Chart.js-powered frontend enables intuitive visualization of trip metrics and trends.
- **Feedback Module:** Generates performance and safety suggestions, exportable as CSV or PDF.

B. Data Collection and Preprocessing

The analytical core comprises seven heterogeneous datasets, covering diverse vehicle types and driving styles. The data pipeline includes:

- **Cleaning:** Detects and removes missing values, inconsistencies, and outliers.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

- **Harmonization:** Standardizes schema across datasets, unifying variable names and formats.
- **Feature Standardization:** Applies z-score normalization to align different value scales (e.g., speed vs. acceleration).
- **Centralized Storage:** All cleaned data is stored in an SQLite database optimized for analytical queries and dashboard rendering.

C. Feature Engineering and Trip Labeling

A handpicked set of behavioral features was derived from the cleaned data. For each trip, thirteen aggregated features are computed:

- Speed-related features: average speed, maximum speed, speed standard deviation;
- Engine-related features: average RPM, maximum RPM, RPM standard deviation;
- Control-related features: average throttle position, maximum throttle position, average engine load;
- Acceleration-related features: average acceleration, acceleration standard deviation;
- Event-based features: number of brake events, number of speed changes.

Additional signals such as steering angle, angular velocity, tire pressure, and GPS coordinates (where available) are used only for visualization and qualitative analysis and are not included in the machine learning feature vector.

Label Assignment: Each trip in our project is labeled as *Safe*, *Moderate*, or *Aggressive* using a hybrid methodology:

- Domain heuristics (e.g., thresholds for RPM, braking events, speed variability);
- Inspection of exceptional or edge cases;
- Optimization of the label boundaries through clustering analysis.

After computing the trip-level features, unsupervised K-Means clustering with $k = 3$ is applied to selected behavior-related features, including speed standard deviation, RPM standard deviation, average throttle position, acceleration standard deviation, number of brake events, and number of speed changes. The resulting clusters are interpreted by analyzing their centroid characteristics. One cluster demonstrates low variability and a small number of events, corresponding to *Safe* driving behavior. A second cluster exhibits moderate variability and event frequency, representing *Moderate* driving behavior. The third cluster shows high variability and frequent events, indicative of *Aggressive* driving behavior.

These cluster-derived behavioral categories are subsequently treated as pseudo-ground-truth labels and used as target classes for training the supervised classification models.

D. Model Development and Evaluation

Multiple ML models were benchmarked, including Random Forest, SVM, Logistic Regression, k-NN, Gradient Boosting, Decision Tree, and Naive Bayes.

- The dataset was split using an 80/20 stratified train-test split.
- 5-fold cross-validation was performed on each model.
- Evaluation metrics included accuracy, precision, recall, and F1-score. Confusion matrices were also analyzed.

IV. System Architecture

The architecture of the Vehicle Trip Analysis Dashboard is designed to be modular, scalable, and deployable on lightweight infrastructure. The system is divided into three primary layers, each responsible for specific functions in the data processing pipeline from sensing to visualization.

A. Sensing Layer

This is the data collection layer that gathers raw vehicle and location data from multiple sources. The sensing layer forms the foundation of the entire system by providing comprehensive trip data acquisition capabilities.

OBD-II Sensor: Collects critical vehicle diagnostics including engine RPM, vehicle speed, fuel consumption rates, throttle position, brake events, and engine load parameters.

GPS Data: Provides distance and duration information where available.

B. Network Layer

This layer is responsible for communication (where applicable), data storage, and machine learning processing. It serves as the central processing hub that transforms raw sensor data into actionable insights.

Flask Backend Framework: A lightweight Python web framework that receives, processes, and manages incoming sensor data or uploaded logs. Flask handles API endpoints, data validation, user authentication, and serves as the primary interface between the sensing layer and data processing components.

SQLite Database: Lightweight, embedded database system used to store comprehensive trip data, user information, and historical driving patterns.

K-Means + Logistic Regression: K-Means clusters trip records into behavioral groups, which are then used as labels to train the Logistic Regression classifier that predicts driving behavior categories (Safe, Moderate, Aggressive).

C. Processing Layer

This is the visualization and user interface layer that presents processed data and machine learning predictions to end users in an intuitive and actionable format.

Web Dashboard: Interactive web-based interface built using HTML5, CSS3, JavaScript, and Chart.js that displays comprehensive trip statistics, behavior scores, and ML predictions. The dashboard provides post-trip visualization of driving patterns through dynamic charts, graphs, and tabular data presentations.

The dashboard includes multiple visualization components:

- Speed vs. time analysis charts
- RPM variation patterns
- Fuel consumption trends
- Braking event distributions
- Behavior classification summaries

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

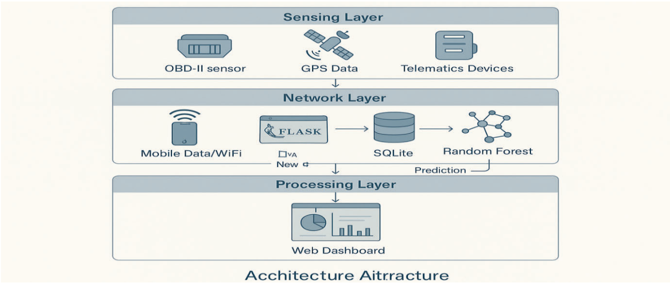

Fig. 1. Three-Layer System Architecture for Vehicle Trip Analysis Dashboard

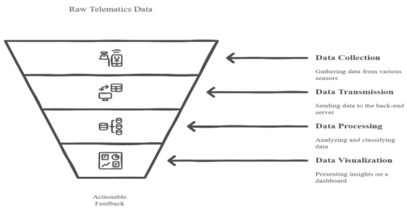

Fig. 2. Pipeline

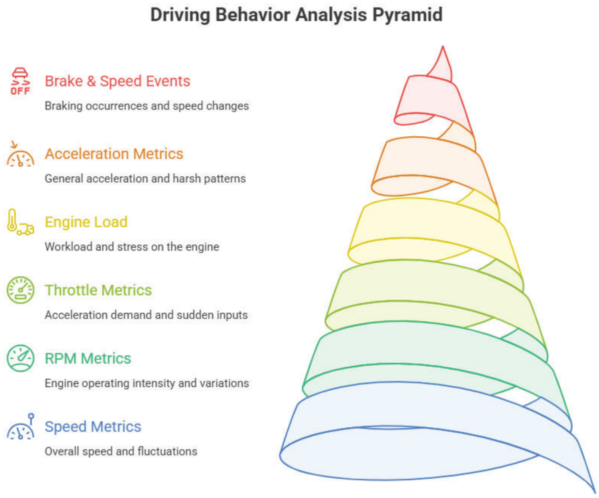

Fig. 3. Driving Behavior Parameters

- Personalized improvement recommendations

The current implementation operates in post-trip mode on pre-recorded logs, while the architecture is designed to be extendable to near real-time ingestion in future work.

V. Dataset and Feature Engineering

A. *Dataset Collection and Preprocessing*

Source Diversity: Data is a strong amalgamation of seven heterogeneous telematics sources (OBD-II sensor logs, smartphone sensor exports, and open driving behavior datasets) collected and curated over time [15]–[21]. Such heterogeneity guarantees a variety of vehicle types, road conditions, and driver behaviours, which supports model generalization.

Preprocessing Pipeline: Before feeding the raw data into the pipeline, all sources are processed through a common sequence:

- **Missing Data Handling:** Systematic treatment of missing values; records that cannot be recovered are discarded.
- **Deduplication & Type Enforcement:** Automatic removal of duplicate and non-numeric artifacts.
- **Schema Normalization:** Use of fuzzy column matching logic to harmonize disparate datasets into a unified schema without manual intervention.
- **Feature Standardization:** Z-score normalization applied to all key numeric features used in ML.
- **Storage:** Cleaned, consolidated dataset stored in a modular SQLite database.

B. *Feature Engineering*

For each trip, thirteen aggregated features are computed:

- Speed-related: average speed, maximum speed, speed standard deviation.
- Engine-related: average RPM, maximum RPM, RPM standard deviation.
- Control-related: average throttle position, maximum throttle position, average engine load.
- Acceleration-related: average acceleration, acceleration standard deviation.
- Event-based: number of brake events, number of speed changes.

Additional signals (steering angle, angular velocity, tire pressure, GPS) are only used for visualization and qualitative analysis, not for ML training.

Label Assignment: Each trip is labeled as *Safe*, *Moderate*, or *Aggressive* using:

- Domain heuristics for thresholds;
- Edge-case inspection;
- K-Means clustering (k=3) to refine boundaries.

C. *Feature Importance*

Analysis of the Logistic Regression model coefficients indicated that speed standard deviation, maximum RPM, acceleration standard deviation, brake events, and RPM standard deviation are the most influential features for behavior classification. Aggregate features such as average speed and average throttle also contribute, but variability and event-based features play a stronger role.

D. *Dataset Integrity and Expandability*

Integrity: The ETL pipeline performs strict type checking, numeric range validation, and consistency checks to ensure scientific reliability.

Expandability: The structure is modular and allows incorporation of future data sources (new OBD-II fields, additional telematics formats) with minimal code changes.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

TABLE I
KEY FEATURES USED FOR DRIVING BEHAVIOR CLASSIFICATION

Feature Name	Type	Rationale / Significance
Average Speed (km/h)	Numeric	Indicates overall speed level and driving smoothness
Maximum Speed (km/h)	Numeric	Helps identify over-speeding and aggressive driving
Speed Standard Deviation	Numeric	Captures variability; high values suggest erratic driving
Average RPM	Numeric	Reflects typical engine operating intensity
Maximum RPM	Numeric	High values indicate aggressive acceleration/gear usage
RPM Standard Deviation	Numeric	Measures engine speed fluctuation
Average Throttle (%)	Numeric	Indicates average acceleration demand
Maximum Throttle (%)	Numeric	Captures strong acceleration bursts
Average Engine Load (%)	Numeric	Reflects engine workload and stress
Average Acceleration	Numeric	Overall acceleration tendency
Acceleration Std Dev	Numeric	Captures harsh acceleration/braking pattern intensity
Brake Events	Integer	Number of braking events; higher counts reflect harsher style
Speed Changes	Integer	Number of speed change events; indicates stability of driving

VI. MACHINE LEARNING MODEL DEVELOPMENT AND EVALUATION

A. Model Selection and Comparative Analysis

Rationale behind Algorithm choice: Several models were benchmarked, including Random Forest, SVM, XGBoost, and Logistic Regression. Logistic Regression was ultimately selected as the deployed model, as it achieved the highest accuracy (90.0%) and F1-score (0.9010) on the trip-level classification task while remaining lightweight and interpretable.

Comparison of Models:

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.9000	0.9076	0.9000	0.9010
SVM (RBF Kernel)	0.8667	0.8694	0.8667	0.8665
Random Forest	0.6667	0.6746	0.6667	0.6650
XGBoost	0.6667	0.6628	0.6667	0.6620

TABLE II
MODEL PERFORMANCE COMPARISON

B. Training Protocol and Methodology

Data Splitting and Validation:

- **Split Strategy:** 80/20 stratified split ensuring balanced class representation across Safe, Moderate, and Aggressive categories.
- **Cross-Validation:** 5-fold stratified cross-validation for robust performance estimation.
- **Feature Integrity:** Only validated "leak-free" features were used, excluding variables directly involved in scoring calculations to prevent data leakage.

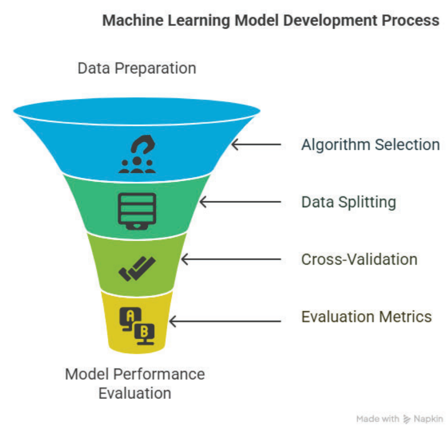

Fig. 4. Model Process

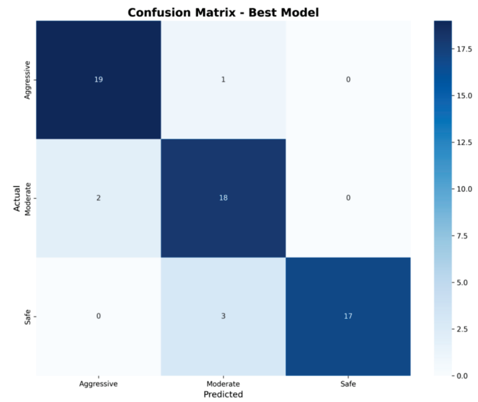

Fig. 5. Example Confusion Matrix

C. Feature Engineering and Importance Analysis

The model employs the trip-level features described earlier, grouped conceptually as:

- **Driving Dynamics:** avg_speed_kmph, max_speed, speed_std, trip_duration (where available), distance_km;
- **Vehicle Performance:** avg_rpm, max_rpm, rpm_std, avg_engine_load, fuel_consumed (for health/efficiency rules);
- **Safety Indicators:** brake_events, accel_std, speed_changes.

Post-training analysis of Logistic Regression coefficients shows variability and event-based features as dominant.

D. Model Performance and Validation

Classification Performance: The final Logistic Regression model achieved:

- Overall Accuracy: 90.0%;
- Weighted F1-Score: 0.9010;
- Weighted Precision: 0.9076;
- Weighted Recall: 0.9000.

Random Forest and XGBoost did not exceed 66.7% accuracy in this configuration and were not selected for deployment.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

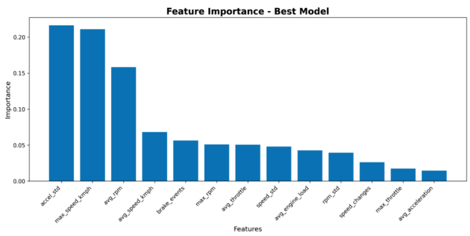

Fig. 6. Feature Importance Illustration

E. Hybrid Scoring System Integration

Dual Approach: The system employs a hybrid methodology combining:

1) **Rule-Based Scoring:** Interpretable feedback using weighted normalization of driving metrics (e.g., speed, RPM, brake events) to produce a 0–100 risk/health score.
2) **ML Classification:** Logistic Regression model for robust behaviour categorization into Safe, Moderate, or Aggressive.

Rule-Based Algorithm (template):

```
score = (
    (avg_speed / 100) * 0.10 +
    ((6000 - max_rpm) / 6000) * 0.10 +
    ((15 - brake_events) / 15) * 0.10 +
    ...
) * 100
```

F. Deployment and System Integration

Production Implementation:

- **Model Serialization:** joblib-based model persistence for Flask backend integration.
- **Fast Post-Trip Inference:** Sub-second prediction latency (0.18 seconds average).
- **Scalability:** Modular architecture supporting easy model replacement and retraining.
- **API Integration:** RESTful endpoints for dashboard and external system connectivity.

Quality Assurance: Thorough testing produced 100% pass rate across 23 functional test scenarios and maintained strong performance under concurrent load.

G. Validation and Reproducibility

Scientific Rigor:

- Random seed controls and stored dataset splits for reproducibility.
- Systematic data validation to protect against corrupt and duplicate information.
- Stratified sampling to ensure representative training/testing distributions.
- Exclusion of target-derived variables to prevent feature leakage.

This end-to-end machine learning pipeline provides an efficient and interpretable solution for trip-wise driving behavior classification suitable for safety-sensitive applications.

VII. Results and Discussion

This section presents the empirical evaluation of the Vehicle Trip Analysis Dashboard, including functionality assessment, ML inference performance, visualization capabilities, and user experience feedback.

A. Functional Testing Results

Core features (user management, data processing, dashboard functionality, trip analysis, report generation) were verified via comprehensive functional test cases.

TABLE III
Functional Testing Summary

Test Case	Expected Result	Outcome
User Registration/Login	Successful user creation and secure session management	Pass
Trip Summary Display	Trip data loaded correctly and sorted by date	Pass
Trip Detail View & Chart Rendering	Interactive graphs for speed, RPM, and fuel consumption	Pass
ML Scoring and Classification	Accurate Safe/Moderate/Aggressive output with numerical score	Pass
Export to CSV/PDF	Files downloaded correctly with complete data	Pass
Mobile Responsiveness	Dashboard functionality on mobile devices	Pass

B. Performance Testing Analysis

To evaluate system efficiency under realistic conditions, performance testing was conducted with 10 concurrent users and high-frequency dashboard interactions.

TABLE IV
System Performance Metrics

Metric	Target Threshold	Observed Value	Status
Dashboard Load Time	< 2 seconds	1.4 seconds	Pass
Trip Detail Analysis	< 1 second	0.45 seconds	Pass
ML Inference Time	< 1 second	0.18 seconds	Pass
Chart Rendering	< 1 second	0.4 seconds	Pass
Database Query Response	< 0.5 seconds	0.23 seconds	Pass

C. Visualization Insights and Capabilities

The Vehicle Trip Analysis Dashboard leverages Chart.js for interactive rendering of driving data in two primary modes: (A) Combined Trip Analytics and (B) Individual Trip Analysis.

1) Combined Trip Analytics Dashboard: The "All Trips" dashboard aggregates metrics from multiple sessions to reveal trends over time:

- **Performance Metrics:** Distance, average speed, and peak speed trends.
- **Efficiency and Load:** Fuel consumption patterns, maximum RPM, and engine load over trips.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

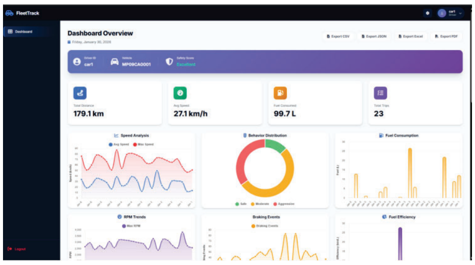

Fig. 7. All Trips Visualization

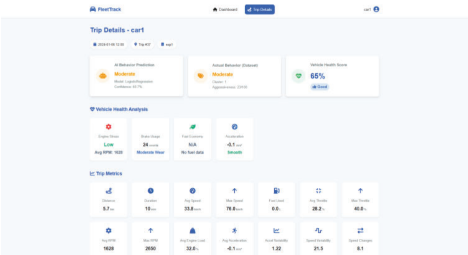

Fig. 8. Trip Detail

- **Safety Indicators:** Braking event counts and acceleration variability.
- **Vehicle Health Metrics:** Where available, additional indicators such as tire pressure are shown for context.

2) Individual Trip Analysis Dashboard: For a specific trip, users see:

Trip Performance Charts:

- Speed and RPM time-series;
- Braking intensity and frequency;
- Engine performance (load, throttle, fuel usage);
- Fuel efficiency curve (e.g., km/L);
- Optional steering dynamics (visual only).

Trip Summary Metrics:

- **Driving Label & Score:** Logistic Regression-based behavior label (Safe/Moderate/Aggressive) plus a rule-based 0–100 risk score;
- **Trip Stats:** Distance, duration, average/max speed and RPM, acceleration, brake events, throttle usage;
- **System Status:** Engine load and optional sensor-based indicators;
- **Health Feedback:** Maintenance alerts and safety recommendations based on stress indicators.

D. ML Classification Case Study

Example trip:

- Average Speed: 68 km/h
- Maximum RPM: 5400
- Brake Events: 7
- Fuel Consumed: 6.4 liters

System output:

- **Behavior Category:** Moderate
- **Quantitative Score:** 71.2 / 100

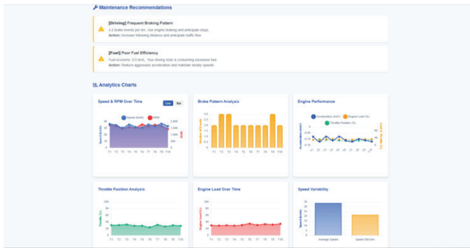

Fig. 9. Single Trip Visualization

- **Feedback:** "Maintain steadier RPM and reduce unnecessary braking to improve safety and fuel economy."

E. User Experience and Usability Evaluation

A usability evaluation with 5 graduate students and 2 faculty advisors gave:

TABLE V
USABILITY FEEDBACK SUMMARY

Evaluation Criteria	Average Rating (/5)
Navigation Flow and Intuitiveness	4.8
Visual Appeal and Design Quality	4.6
Feedback Relevance and Actionability	4.7
Mobile Responsiveness	4.5
Overall User Satisfaction	4.8
Learning Curve and Ease of Use	4.6

Users appreciated the interactive charts and personalized ML-based feedback. Suggestions included more granular historical filtering and optional voice-based feedback in future versions.

VIII. CONCLUSION AND FUTURE WORK

This work presents an extensive Driving Behavior Analysis System aimed at improving vehicle safety and driving performance through post-trip telematics analysis. The system harmonizes data from seven heterogeneous sources, automates data cleaning and schema normalization, and computes interpretable trip-level features.

A modular machine learning pipeline uses K-Means clustering (k=3) to derive behavioral labels and a Logistic Regression classifier to perform trip-wise behavior prediction. Logistic Regression achieved 90.0% accuracy and 0.901 weighted F1-score, outperforming SVM (86.7%) and ensemble models such as Random Forest and XGBoost (66.7% each). This demonstrates that with carefully engineered trip-level features, a simple and interpretable linear model can outperform more complex classifiers on this task.

An interactive dashboard built with Flask and Chart.js provides users with post-trip insights, including behavior labels, risk scores, detailed charts, and vehicle health recommendations. Functional, performance, and usability testing indicate that the system is reliable, responsive, and user-friendly.

Future Work:

- Extend the architecture to near real-time ingestion for in-trip feedback while preserving interpretability.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 02 , February - 2026

- Integrate contextual data such as traffic, weather, and road type to better separate driver behavior from environmental factors.
- Explore temporal sequence models (e.g., LSTMs, Transformers) to capture within-trip dynamics and short aggressive events.
- Expand the dataset to cover more vehicle types, demographics, and geographical regions for improved generalization.
- Incorporate explainable AI techniques (e.g., SHAP) to provide per-trip explanations of model decisions.

References

[1] T. Board, A. Smith, and E. Johnson, "Exploring the impact of driving behavior on safety using traditional and sensor-based data collection," *Journal of Vehicle Safety*, vol. 32, no. 2, pp. 212–220, 2018.

[2] H. Singh, S. Kumar, and M. Patel, "Machine learning-based classification of driving patterns using GPS and accelerometer data," *IEEE Trans. Intell. Veh.*, vol. 6, no. 1, pp. 88–97, 2021.

[3] U. Fugiglando, P. Santi, S. Milardo, *et al.*, "Driving behavior analysis through CAN bus data in an uncontrolled environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 1178–1187, Mar. 2019.

[4] M. Rizbood, A. Nair, and R. Joshi, "Smartphone sensor-based driving behavior analysis for safety and insurance applications," *Procedia Comput. Sci.*, vol. 170, pp. 1009–1014, 2020.

[5] P. Garefalakis, M. Chliogiorgou, and G. Tzimas, "Comparison of machine learning models for driving behavior classification," *Procedia Comput. Sci.*, vol. 170, pp. 110–117, 2020.

[6] A. Jain and A. Mittal, "Fuel-efficient driving style detection using deep learning techniques," *IEEE Access*, vol. 8, pp. 131 793–131 803, 2020.

[7] S. Kwon, Y. Kim, and H. Lee, "Driving style classification using a deep neural network," *Sensors*, vol. 19, no. 20, p. 4565, 2019.

[8] R. Choudhary and S. Meena, "A survey on AI-based driver assistance systems," *J. Transp. Technol.*, vol. 11, pp. 85–102, 2021.

[9] A. Kashevnik, I. Lashkov, and N. Shilov, "Real-time driving behavior analysis for autonomous vehicle control," *Procedia Comput. Sci.*, vol. 169, pp. 150–157, 2020.

[10] J. S. Hickman, C. Perez, and M. Brewer, "Optimizing driving strategies for autonomous vehicles using reinforcement learning," *Transp. Res. Part C*, vol. 105, pp. 456–472, 2019.

[11] F. Liao, Y. Dong, and F. Wang, "Human–AI interaction in driving assistance systems: A safety perspective," *IEEE Trans. Syst., Man, Cybern.*, vol. 51, no. 6, pp. 3029–3038, Jun. 2021.

[12] R. Zylius, M. Markevicius, and L. Sakalauskas, "Ethical concerns in AI-driven driving behavior analysis: Privacy and decision-making," *Artif. Intell. Ethics*, vol. 2, pp. 160–170, 2021.

[13] Frotcom, "Fleet intelligence dashboard: Impact on driver behavior and safety," Frotcom Int. White Paper, 2022.

[14] A. Arumugam and S. Rajasekaran, "An overview of usage-based insurance: From telematics to manage-how-you-drive models," *IEEE Access*, vol. 8, pp. 85 070–85 084, 2020.

[15] C. Barreto, "OBD-II datasets (obdii-ds3)," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/cephasax/obdii-ds3

[16] E. da Silva Neto, "obd2data," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/eron93br/obd2data

[17] R. Gagrani, "OBD-II dataset," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/rishitagagrani/obd-ii-dataset

[18] E. Neto, "carOBD: An OBD-II database for Toyota Etios 2014 vehicle," GitHub, 2020. [Online]. Available: https://github.com/eron93br/carOBD

[19] P. Sekhar, "VehicalDiagnosticAlgo," GitHub, 2021. [Online]. Available: https://github.com/prithvisekhar/VehicalDiagnosticAlgo

[20] M. Bhele, "vehicletelematics," GitHub, 2023. [Online]. Available: https://github.com/mukul-bhele/vehicletelematics

[21] M. Weber, "Automotive OBD-II Dataset," KIT RADAR Repository, 2023. [Online]. Available: https://radar.kit.edu/radar/en/dataset/bCtGxdTklQlfQcAq